Министерство образования и науки Украины

Харьковский национальный университет радиоэлектроники

Кафедра БИТ

Отчет

 По лабораторной работе по ПБОЗИ

Тема: «Операции с элементами GF(2*m*) в полиномиальном базисе»

Уровень А

Выполнили:  Проверил: ст.гр БИКС-14-1 Мельникова О. А.

Д. Тхао

Харьков 2017

#include <iostream>

#include "MMATH.H"

#include "PolynomsLib\_Bin\_1.H"

#include "PolynomsLib\_IO.h"

#include "AddDblPointsEC\_LopezDahab.h"

using namespace std;

void printPoint(Point P) {

if (P.x[0] == 1 && P.x[1] == 0 && P.y[0] == 1 && P.y[1] == 0)

cout << " I ";

else printf(" (%x, %x) ", P.x[1], P.y[1], P.z[1]);

}

void OrdPointEC(Point P, Point Q, EC Curve, unsigned int &i)

{

Point R = { 0 };

AddDblPointsEC\_LopDak(P, Q, Curve, R);

i++;

if (R.x[0] != 1 || R.x[1] != 0 || R.y[0] != 1 || R.y[1] != 0)

OrdPointEC(P, R, Curve, i);

}

bool IsOnCurve(Point P, EC Curve)

{

M\_LONG l, r, tmp, fin;

sqrmod\_pol(P.y, Curve.f, l);

mulmod\_pol(P.x, P.y, Curve.f, tmp);

add\_pol(l, tmp, l);

rem\_pol(l, Curve.f);

sqrmod\_pol(P.x, Curve.f, tmp);

mulmod\_pol(tmp, P.x, Curve.f, fin);

mulmod\_pol(tmp, Curve.a, Curve.f, r);

add\_pol(fin, r, r);

add\_pol(r, Curve.b, r);

rem\_pol(r, Curve.f);

if (m\_cmp(l, r) == 0) return 1;

return 0;

}

int main()

{

unsigned int x, y, i = 0;

Point set[20], R = { 0 };

EC Curve = { {1,2}, {1,0xe}, {1,0x13} };

cout << "\t |";

for (x = 0; x <= 0xf; x++)

for (y = 0; y <= 0xf; y++) {

set[i].x[0] = 1; set[i].x[1] = x;

set[i].y[0] = 1; set[i].y[1] = y;

if (IsOnCurve(set[i], Curve)) {

set[i].z[0] = 1; set[i].z[1] = 1;

set[i].ord = 1;

printPoint(set[i]);

OrdPointEC(set[i], set[i], Curve, set[i].ord);

i++;

}

}

cout << endl << "--------------------------------------------------";

cout << "-----------------------------------------------------------\n";

for (int j = 0; j < i; j++) {

printPoint(set[j]); cout << '|';

for (int k = 0; k < i; k++) {

AddDblPointsEC\_LopDak(set[j], set[k], Curve, R);

printPoint(R);

}

cout << endl;

}

cout << "--------------------------------------------------------";

cout << "-----------------------------------------------------\n";

cout << "OrdPoint |";

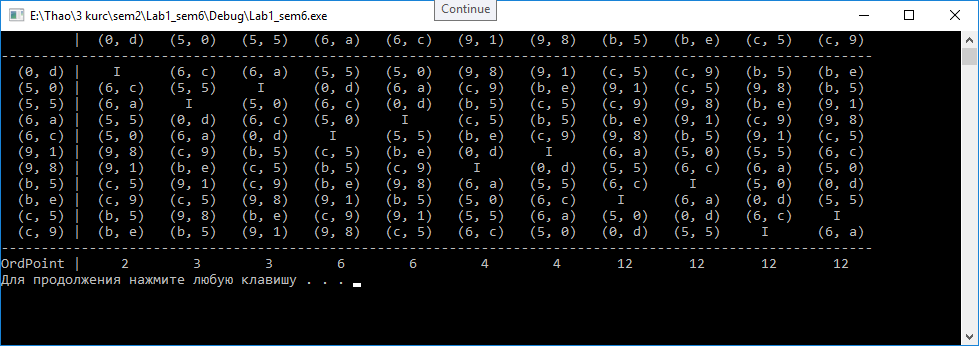
for (int j = 0; j < i; j++) printf(" %2d ", set[j].ord);

cout << endl;

system("pause");

return 0;

}



Файл AddDblPointsEC\_LopezDahab.cpp:

#include "MMATH.H"

#include "PolynomsLib\_Bin\_1.h"

struct EC {

M\_LONG a, b, f;

int OrdEC;

};

struct Point {

M\_LONG x, y, z;

int ord;

};

int AddDblPointsEC\_LopDak(Point &P, Point &Q, EC &Curve, Point &R)

{

M\_LONG tmp1, tmp2, tmp3;

if (P.z[0] == 1 && P.z[1] == 0) {

m\_copy(R.x, Q.x); m\_copy(R.y, Q.y); m\_copy(R.z, Q.z);

return 0;

}

if (Q.z[0] == 1 && Q.z[1] == 0) {

m\_copy(R.x, P.x); m\_copy(R.y, P.y); m\_copy(R.z, P.z);

return 0;

}

if (m\_cmp(P.x, Q.x) == 0) {

if (m\_cmp(P.y, Q.y) != 0 || (P.x[0] == 1 && P.x[1] == 0)) {

R.x[0] = 1; R.x[1] = 0;

R.y[0] = 1; R.y[1] = 0;

R.z[0] = 1; R.z[1] = 0;

return 0;

}

sqrmod\_pol(P.x, Curve.f, tmp1); //tmp1 = x^2

sqrmod\_pol(P.z, Curve.f, tmp2); //tmp2 = z^2

mulmod\_pol(tmp1, tmp2, Curve.f, R.z); //z3 = x^2 \* z^2

sqrmod\_pol(tmp1, Curve.f, tmp3); //tmp3 = x^4

sqrmod\_pol(tmp2, Curve.f, tmp1); //tmp1 = z^4

mulmod\_pol(tmp1, Curve.b, Curve.f, tmp2);//tmp2 = z^4 \* b

add\_pol(tmp2, tmp3, R.x); //x3 = x^4 + b\*z^4

mulmod\_pol(Curve.a, R.z, Curve.f, tmp1); //tmp1 = a \* z3

add\_pol(tmp2, tmp1, tmp3); //tmp3 = a \* z3 + z^4 \* b

sqrmod\_pol(P.y, Curve.f, tmp1); //tmp1 = y^2

add\_pol(tmp1, tmp3, tmp1); //tmp1 = tmp1 + tmp3 = y^2 + a\*z3 + z^4 \*b

mulmod\_pol(tmp1, R.x, Curve.f, tmp3); //tmp3 = x3 \* tmp1

mulmod\_pol(tmp2, R.z, Curve.f, tmp1); //tmp1 = tmp2 \* z3

add\_pol(tmp1, tmp3, R.y); //y3 = tmp1 + tmp3

}

else {

M\_LONG A, B, C, D, E, F, G, tmp1, tmp2, tmp3;

sqrmod\_pol(P.z, Curve.f, tmp1); //tmp1 = z1^2

mulmod\_pol(Q.y, tmp1, Curve.f, A); //A = z1^2 \* y2

add\_pol(A, P.y, A); //A = z1^2 \* y2 + y1

mulmod\_pol(Q.x, P.z, Curve.f, B); //B = x2 \* z1

add\_pol(B, P.x, B); //B = x2 \* z1 + x1

mulmod\_pol(B, P.z, Curve.f, C); //C = B \* z1

mulmod\_pol(tmp1, Curve.a, Curve.f, tmp2); //tmp2 = z1^2 \* a

add\_pol(C, tmp2, tmp3); //tmp3 = z1^2 \* a + C

sqrmod\_pol(B, Curve.f, tmp1); //tmp1 = B^2

mulmod\_pol(tmp3, tmp1, Curve.f, D); //D = B^2 \* tmp3

sqrmod\_pol(C, Curve.f, R.z); //z3 = C^2

mulmod\_pol(A, C, Curve.f, E);

sqrmod\_pol(A, Curve.f, R.x); //tmp1 = A^2

add\_pol(R.x, D, R.x);

add\_pol(R.x, E, R.x);

mulmod\_pol(Q.x, R.z, Curve.f, F);

add\_pol(F, R.x, F);

mulmod\_pol(Q.y, R.z, Curve.f, G);

add\_pol(G, R.x, G);

mulmod\_pol(E, F, Curve.f, tmp1);

mulmod\_pol(R.z, G, Curve.f, R.y);

add\_pol(R.y, tmp1, R.y);

}

inv\_pol(R.z, Curve.f, tmp1);

mulmod\_pol(R.x, tmp1, Curve.f, tmp2);

m\_copy(R.x, tmp2);

sqrmod\_pol(tmp1, Curve.f, tmp2);

mulmod\_pol(R.y, tmp2, Curve.f, tmp1);

m\_copy(R.y, tmp1);

return 0;

}