## ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего профессионального образования

## Национальный исследовательский университет

«Высшая школа экономики»

## Московский институт электроники и математики им. А. Н. Тихонова Департамент прикладной математики

Кафедра компьютерной безопасности

## ОТЧЁТ

по курсовой работе по дисциплине

## «Программирование алгоритмов защиты информации»

Реализация эллиптических кривых в форме Гессе и арифметических операций над точкой

## Выполнил студент группы СКБ171 Миронов Никита Андреевич

Москва 2020

**Цель работы**

Программная реализация эллиптических кривых в форме Гессе, реализация арифметических операций над точкой в поле, реализация и применение тестов для определения корректности выполнения поставленной задачи. Для выполнения работы используются средства библиотеки OpenSSl libcripto.

**Теоретическая часть**

Рассмотрим эллиптическую кривую Гессе в виде кубической формы в проективных координатах

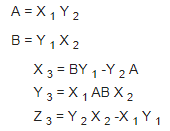


Или в аффинной плоскости и удовлетворяет

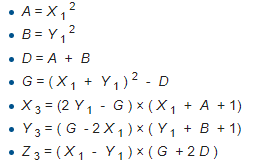
Так же

Кривая в краткой форме Вейерштрасса 

Пусть P 1 = (X 1 : Y 1 : Z 1 ) и P 2 = (X 2 : Y 2 : Z 2 ) - две точки, отличные от . Предполагая, что Z 1 = Z 2 = 1, алгоритм задается следующим образом:



Пусть P = ( X 1 : Y 1 : Z 1 ) - точка, тогда формула удвоения задается следующим образом:



**Практическая часть**

**Используемые функции и реализованные алгоритмы**

Алгоритмы функций были взяты с данного ресурса (https://hyperelliptic.org/EFD/g1p/auto-hessian-standard.html)

Из файла funk.h используются следующие функции:

Алгоритм сложения точек

void addDots(struct Coordinates\* D1, struct Coordinates\* D2, struct Coordinates\* D3, struct Numbers\* crv);

Алгоритм создание точки

void makeDot(struct Coordinates\* dot, char\* X, char\* Y, char\* Z);

Алгоритм для проверки того что функция лежит на кривой

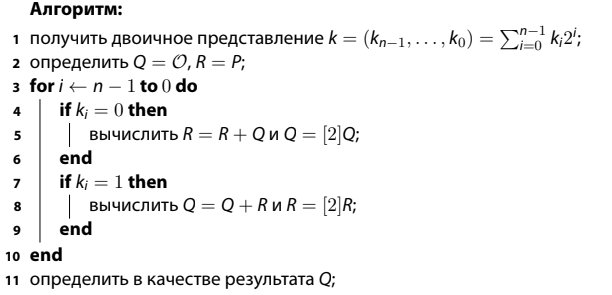
int ifonHesse (struct Coordinates\* D, struct Numbers\* crv);

Алгоритм сравнения двух точек

int Equal(struct Coordinates\* FirstD, struct Coordinates\* SecondD, struct Numbers\* crv);

Алгоритм вычисления кратной точки с помощью алгоритма Монтгомери

void findDot(struct Coordinates\* DK, struct Coordinates\* D, struct Numbers\* crv, BIGNUM\* Num);



Алгоритм перевода в аффинные координаты

void swapAFF(struct Coordinates\* affD, struct Coordinates\* D, struct Numbers\* crv);

**Описание использованных функций из библиотеки OpenSSl libcripto**

Определения, использованных в программе функций, описаны в заголовочном файле bn.h.

Арифметические операции

* int BN\_mod\_add(BIGNUM \*r, const BIGNUM \*a, const BIGNUM \*b, const BIGNUM \*m, BN\_CTX \*ctx);
* int BN\_mod\_sub(BIGNUM \*r, const BIGNUM \*a, const BIGNUM \*b, const BIGNUM \*m, BN\_CTX \*ctx)
* int BN\_mod\_mul(BIGNUM \*r, const BIGNUM \*a, const BIGNUM \*b, const BIGNUM \*m, BN\_CTX \*ctx)
* int BN\_mod\_sqr(BIGNUM \*r, const BIGNUM \*a, const BIGNUM \*m, BN\_CTX\*ctx)
* int BN\_exp(BIGNUM \*r, const BIGNUM \*a, const BIGNUM \*p, BN\_CTX \*ctx)
* int BN\_dec2bn(BIGNUM \*\*a, const char \*str) забор символов из строки
* .int BN\_is\_zero(const BIGNUM \*a) a проверка на 0

Установка библиотеки производилась на Cilon c помощью Cygwin

**Константы**

Константы для работы были взяты из лекции Программирование алгоритмов защиты информации(<https://drive.google.com/drive/u/1/folders/17_F1NM91KR-6HOnUG_pHWxmtlRf7auU1>)

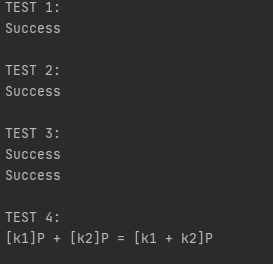
**Тесты**

Тесты располагаются в файле main.c

**Код и результат работы программы**

Код: <https://github.com/h2sapien/pazi_1>

Результат:



**Используемые источники**

Лекции “Программирование алгоритмов защиты информации”. автор курса: А.Ю Нестеренко

<https://hyperelliptic.org/EFD/>

“Hessian Elliptic Curves and Side-Channel Attacks” автор Marc Joye1 and Jean-Jacques Quisquater