Правительство Российской Федерации

Федеральное государственное автономное образовательное   
учреждение высшего образования

Национальный исследовательский университет

«Высшая школа экономики»

Московский институт электроники и математики

Факультет прикладной математики и кибернетики

Кафедра «Компьютерная безопасность»

**КУРСОВАЯ РАБОТА**

по дисциплине «Программирование алгоритмов защиты информации»

Программная реализация алгоритма электронной подписи ECDSA

с использованием библиотеки gmp

**Выполнил(а):** студент группы СКБ172 Попов Ю.Л.

**Проверил:** доцент Нестеренко А. Ю.

**МОСКВА – 2020**

**Оглавление**

[Введение 2](#_Toc54972437)

[Теоретическая часть 4](#_Toc54972438)

[Работа с библиотекой gmp 7](#_Toc54972439)

[Исходные данные 8](#_Toc54972440)

[Структура программы 10](#_Toc54972441)

[Пример вывода программы 11](#_Toc54972442)

[Список использованной литературы 12](#_Toc54972443)

# Введение

Предметом изучения данной курсовой работы является программная реализация стандарта на электронную подпись – ECDSA.

Американский стандарт на электронную подпись FIPS 186 вначале стандартизировал схему электронной подписи Эль-Гамаля, реализуемую в мультипликативной группе конечного простого поля и получившую название DSA (digital signature algorithm). Начиная со второй редакции данного стандарта допускается возможность реализации схемы DSA в группе точек эллиптической кривой, определенной либо над конечным простым полем Fp, либо над конечным простым полем характеристики 2. Такую схему принято называть ECDSA (elliptic curve digital signature algorithm).

Открытые параметры такой схемы аналогичны параметрам ГОСТ Р 34.10-2012, за исключением используемой функции хеширования.

До последнего времени стандарт FIPS 186 предписывал использовать семейство функций SHA, вырабатывающих коды целостности длины I = 160, 256, 384 и 512 бит. Соответственно, изменение длины кода целостности приводит к изменению величины q, определяющей порядок подгруппы, которую порождает точка Р. Мы будем считать, что для q выполнено неравенство 2l > q.

# Теоретическая часть

Для подписывания сообщений необходима пара ключей — открытый и закрытый. При этом закрытый ключ должен быть известен только тому, кто подписывает сообщения, а открытый — любому желающему проверить подлинность сообщения. Также общедоступными являются параметры самого алгоритма.

# Параметры алгоритма

### Выбор [хеш-функции](https://ru.wikipedia.org/wiki/%D0%A5%D0%B5%D1%88%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) H(x) Для использования алгоритма необходимо, чтобы подписываемое сообщение являлось числом. Хеш-функция должна преобразовать любое сообщение в последовательность битов, которые можно потом преобразовать в [число](https://ru.wikipedia.org/wiki/%D0%A7%D0%B8%D1%81%D0%BB%D0%BE).

1. Выбор большого [простого числа](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%81%D1%82%D0%BE%D0%B5_%D1%87%D0%B8%D1%81%D0%BB%D0%BE) q - порядок одной из [циклических](https://ru.wikipedia.org/wiki/%D0%A6%D0%B8%D0%BA%D0%BB%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D0%B3%D1%80%D1%83%D0%BF%D0%BF%D0%B0) [подгрупп](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%B4%D0%B3%D1%80%D1%83%D0%BF%D0%BF%D0%B0) группы точек эллиптической кривой.
2. [Простым числом](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%81%D1%82%D0%BE%D0%B5_%D1%87%D0%B8%D1%81%D0%BB%D0%BE)  p  будем обозначать характеристику поля координат Fp

### Генерирование ключей ECDSA

Для простоты будем рассматривать эллиптические кривые над полем Fp , где Fp-  [конечное](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D0%B5%D1%87%D0%BD%D0%BE%D0%B5_%D0%BF%D0%BE%D0%BB%D0%B5) [простое](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%81%D1%82%D0%BE%D0%B5_%D0%BF%D0%BE%D0%BB%D0%B5) [поле](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D0%B5_(%D0%B0%D0%BB%D0%B3%D0%B5%D0%B1%D1%80%D0%B0)). Причем, если необходимо, конструкцию можно легко адаптировать для эллиптических кривых над другим полем.

Пусть E - эллиптиеская кривая, определенная над Fp , и P - точка простого порядка q кривой E(Fp. Кривая E и точка P являются системными параметрами. Число  p - простое. Каждый пользователь — условно назовём его [Алиса](https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B8%D1%81%D0%B0_%D0%B8_%D0%91%D0%BE%D0%B1) — конструирует свой ключ посредством следующих действий:

1. Выбирает случайное или псевдослучайное целое число x из интервала [1,q-1]
2. Вычисляет произведение (кратное) Q = x\* P.

Открытым ключом пользователя Алисы  A является точка  Q, а закрытым - x.

### Вычисление цифровой подписи

Для того, чтобы подписать какое-либо сообщение, для которого подсчитано значение h хеш-функции  H, пользователь  A должен сделать следующее:

1. Выбрать случайное целое число k ∈ [1,q-1].
2. Вычислить k\* P = (x1,y1) и положить в r = x1 (mod q), где r получается из целого числа x1 между 0 и (p-1)приведением по модулю q.

*Замечание*: если  r=0, то уравнение подписи s=k-1 h+x\*r (mod q)не зависит от секретного ключа x, и следовательно, (r,s)не подходит в качестве цифровой подписи. Значит, в случае r = 0 необходимо вернуться к шагу 1.

1. Вычислить k-1 (mod q) и положить s=k-1(h + x\*r)(mod q), где h  - значение хеш-функции подписываемого сообщения.

Подписью для сообщения является пара целых чисел (r, s){\displaystyle (r,s)}

### Проверка цифровой подписи

Для того, чтобы проверить подпись пользователя Алисы (r,s) на сообщение, пользователь Боб B должен сделать следующее:

1. Получить подтвержденную копию открытого ключа Q пользователя А;
2. Проверить, что числа r и s являются целыми числами из интервала [1,q-1] и вычислить значение хеш-функции h от сообщения;
3. Вычислить u1= s-1 mod q и  u2 = s-1 r (mod q)
4. Вычислить  u1 P+u2 Q = (x0,y0), и относительно x0, как целого числа между 0 и (p-1), положить v=x0 mod q;
5. Принять подпись, тогда и только тогда, когда v=r

Для подтверждения публичного ключа Q нужно проделать следующее:

1. Проверить, что Q не равно О и координаты верны;
2. Проверить, что Q лежит на кривой;
3. Проверить, что qQ = O;

# Установка библиотеки gmp на Ubuntu:

1. Скачиваем архив с официального сайта: <https://gmplib.org/#DOWNLOAD>. Например,  [gmp-6.2.0.tar.xz](https://gmplib.org/download/gmp/gmp-6.2.0.tar.xz)
2. Разархивируем папку и переходим в неё.
3. Выполняем следующие команды:

./configure

make

make check

make install

Подключение библиотеки к проекту в CMakeLists.txt:



Подключение библиотеки в программе:

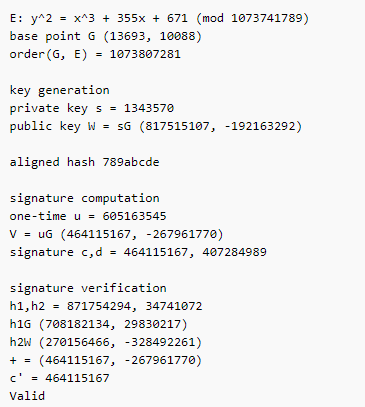
#include <gmp.h>

# Структура программы

Программа состоит из четырех файлов:

1. **main.c** – содержит вычисления, тесты и вывод;
2. **ECDSA.h** – содержит описания структур и все необходимые функции для генерации и верификации электронной подписи
3. **ECDSA.c** – содержит определения функций, описанных в одноименном заголовочном файле.

# Вывод программы

****

# Список использованной литературы

1. Нестеренко А. Ю. – Курс лекций «Методы программной реализации СКЗИ»;
2. O. Billet, M. Joye. The Jacobi model of an elliptic curve and side-channel analysis, proceedings of AAECC-15, Lecture Notes in Computer Science;
3. «Рекомендации по стандартизации. Задание параметров скрученных эллиптических кривых Эдвардса в соответствии с ГОСТ Р 34.10-2012».